Zuul, the Third

Throws Away Any Dirt!

Szymon Datko
szymon.datko@corp.ovh.com

Roman Dobosz
rdobosz@redhat.com

6th November 2019

About us

Szymon Datko

- DevOps & local Bash wizard
- Open Source software lover
- Computer Graphics enthusiast

Roman Dobosz

- Python expert
- 8bit fan
- `emerge -vaNDu world`

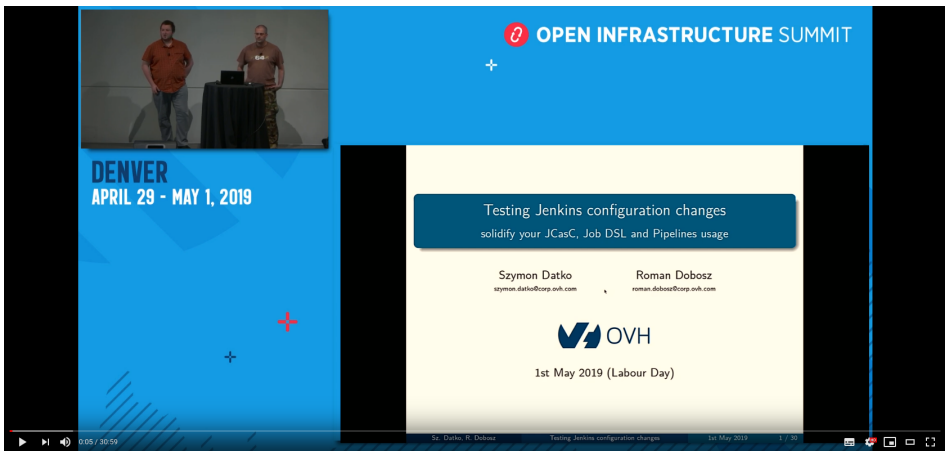
OVHcloud

Red Hat

openstack.

We already talked about Jenkins - three times!

1) Berlin: `https://www.youtube.com/watch?v=T7rD--ZOYRQ`



2) Denver: `https://www.youtube.com/watch?v=nvgeXkE65ac`

and 3) – yesterday on this Open Infrastructure Summit!

OVHcloud

Recently we met a new friend

"Zuule w taxi ... sezamki!"

OVHcloud

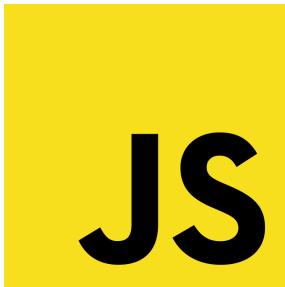
What Zuul **is not**?

There is a name conflict we have to deal with... (OpenStack's Zuul: May 2012)

So, in this presentation we are not talking about:

- *"... a gateway service that provides dynamic routing, monitoring, resiliency, security, and more."*
  - by Netflix,
  - on GitHub since March 2012;

- *"... an easy way to test your javascript in browsers."*
  - by Roman Shtylman / defunctzombie,
  - on GitHub since December 2012.

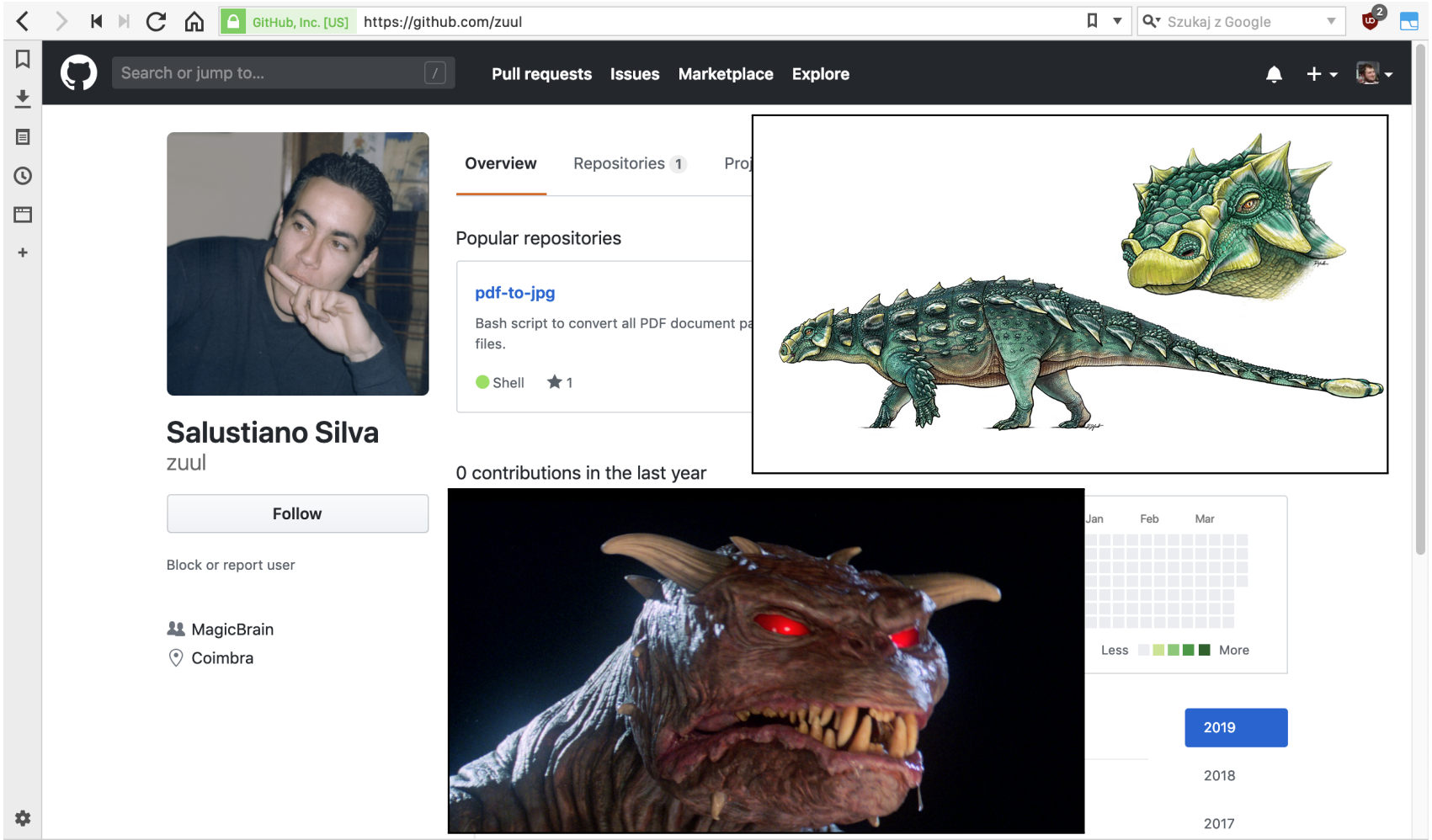


Images sources:
https://onlyfullstack.blogspot.com/2018/09/microservices-with-zuul-gateway29.html,
https://commons.wikimedia.org/wiki/File:Unofficial_JavaScript_logo_2.svg.

OVHcloud

What Zuul **is not**?

Also it is neither an username, nor dinosaur *Zuul crurivastator*.



Images sources:
https://ghostbusters.fandom.com/wiki/Zuul,
https://www.sciencemag.org/news/2017/05/demon-faced-dinosaur-named-after-ghostbusters-baddie.

OVHcloud

What Zuul is?

- Mainly: a project gating system.
- Currently a continuous integration, delivery and deployment system.
- Drives one of the largest CI system in the open source world – OpenStack!
- Support for inter-projects dependencies.

leboncoin BMW GoDaddy openlab openstack. VOLVO RDO Software Factory tungstenfabric packet

Sources:
https://zuul-ci.org/users.html,
https://sdtimes.com/cicd/cicd-platform-zuul-version-3-released/.

OVHcloud

A bit of history...

"Gating is a process where every change, after passing code review, is automatically tested and merged only if it passes the test suite."

James Blair, "How OpenStack Improves Code Quality with Project Gating and Zuul"

- $\sim$ 2010:
  - OpenStack is born; Jenkins used for tests and gating.
- $\sim$ 2012:
  - Zuul comes to life as a coordinator of Jenkins jobs.
  - Goal: parallelize the serial testing (speculative execution).
- $\sim$ 2016:
  - first ideas of replacing Jenkins with Ansible-based execution system.
- $\sim$ 2018:
  - **Zuul v3** finally released!

OVHcloud

What Zuul can do?

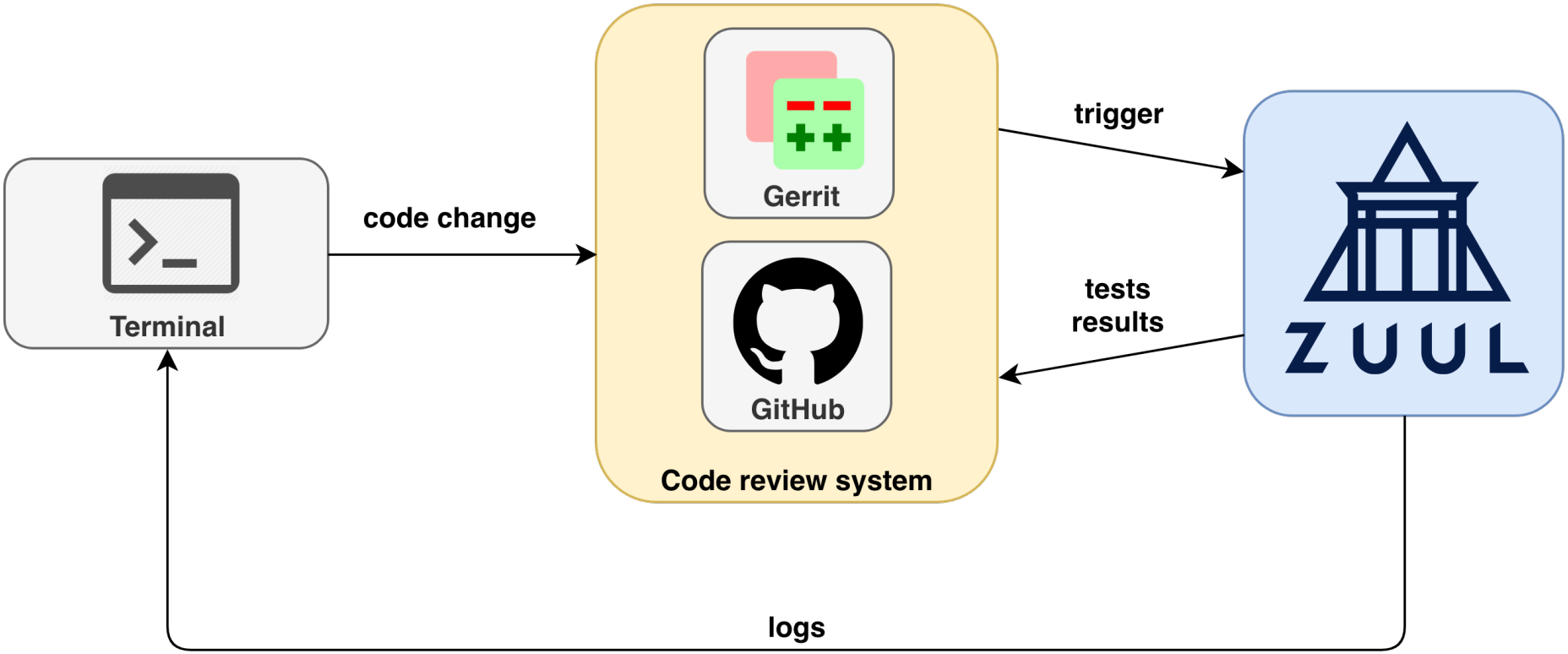
Key features:

- jobs defined as pipelines (YAML files) in git repository,
- Ansible-based executor – launches your jobs anywhere,
- integrates with Gerrit and GitHub systems,
- cross-project and cross-repository-system dependencies mechanism,
- speculative execution for fast and safe automated merging of code,
- pretty scalable* architecture.

* - except for Scheduler; we will come back to this later.

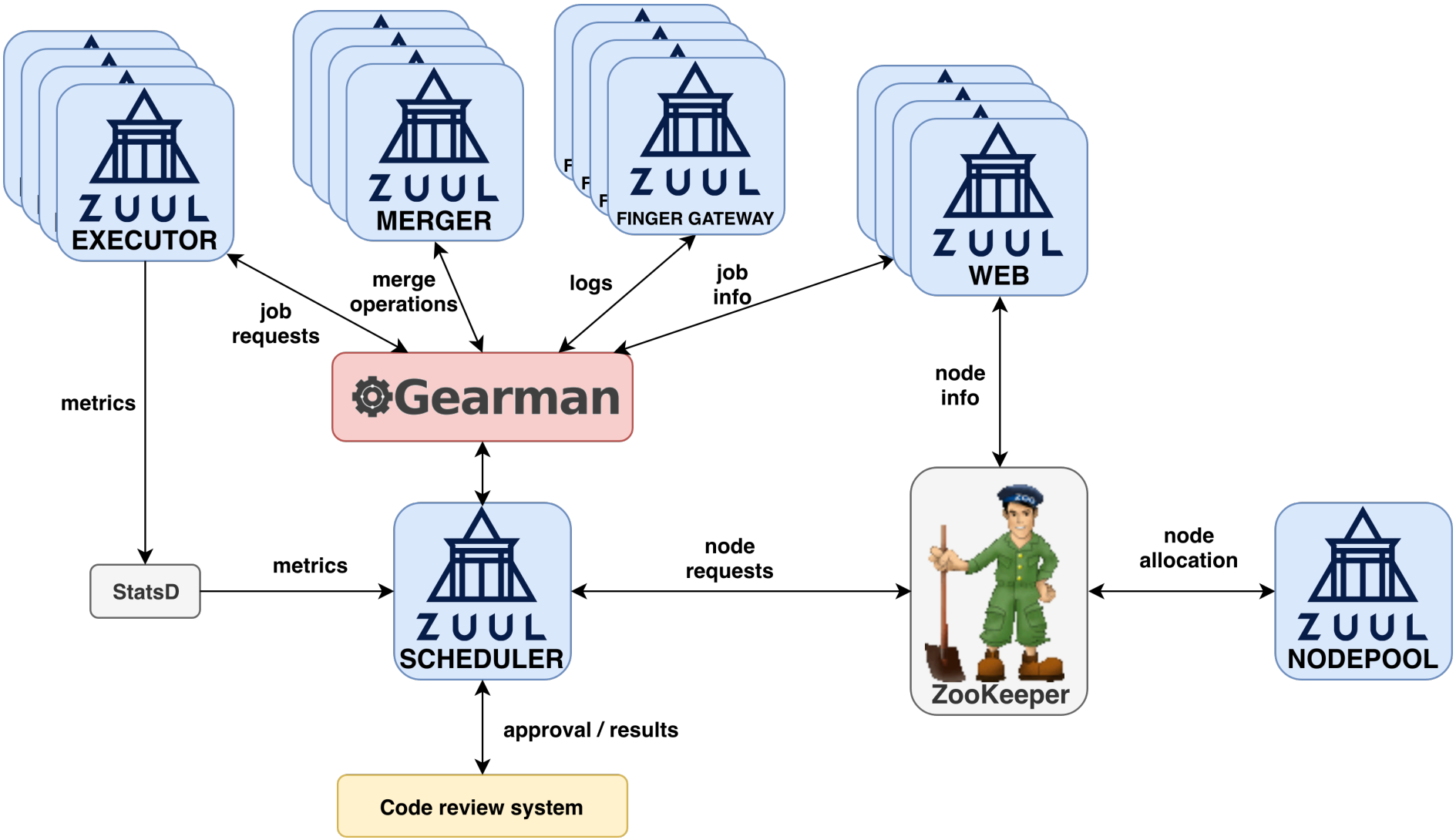
OVHcloud

Zuul in typical environment



OVHcloud

System's architecture



OVHcloud

System's architecture ⇒ Zuul Scheduler

- Primary decision component of Zuul.
- Currently not scalable at all (one at a time!).
- Utilizes Gearman server for communication:
  - Zuul includes own implementation of Gearman,
  - external Gearman server can be used,
  - forcing SSL is strongly recommended.
- Must be connected to ZooKeeper to request nodes – however, does not connect to them!
- Receives events from code review system, then enqueues into pipelines and distributes jobs.

OVHcloud

System's architecture ⇒ Zuul Executor

- Responsible for running jobs.
- At the start of each job:
  - prepares the environment, using Ansible Roles,
  - checkouts the proper projects/branches,
  - creates Ansible inventory file.
- Must be able to connect to:
  - Gearman server,
  - Code Review System,
  - hosts provided by Nodepool.
- Contains also the Merger capabilities.
- Secures contexts using `bubblewrap`

OVHcloud

System's architecture ⇒ Zuul Merger

- Optional component:
  - not necessary in small deployments,
  - highly recommended for large installations.
- Exists to reduce the load on Zuul Executors.
- Performs a lof of `git` operations (speculative merges), which can be time consuming.
- Needs an access to:
  - Gearman server,
  - Code Review System.

OVHcloud

System's architecture ⇒ Zuul Finger Gateway

- Listens for Finger protocol requests.
- Finds which Executor is running a desired build.
- Returns a log stream from executor.
- Needs an access to:
  - Gearman server,
  - console streaming port on the Executor.

OVHcloud

System's architecture ⇒ Zuul Web

- Web-based interface to interact with Zuul.
- Simple, single-process application.
- Written in ReactJS.
- Provides:
  - websockets for live log streaming,
  - the REST API,
  - HTML dashboard (GUI).
- Must have a connection to Gearman.

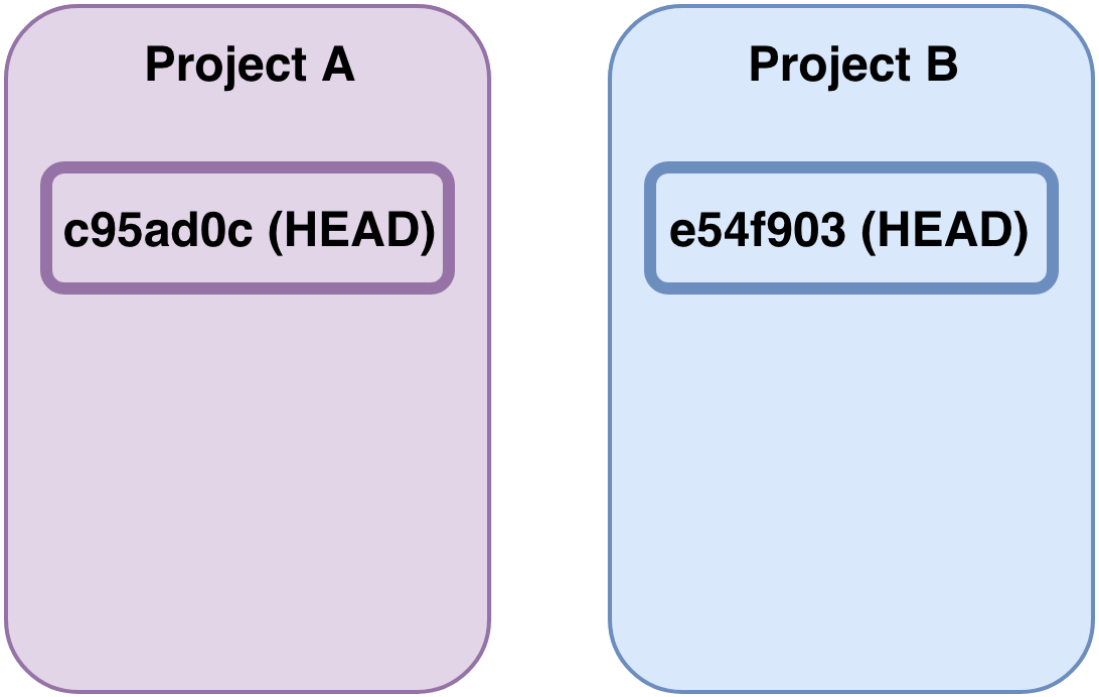
OVHcloud

System's architecture ⇒ Nodepool

- Manages hosts nodes for tests executors.
- Actually, a set of many tools.
- May launch single-use nodes on demand.
- Capable of caching and provisioning pre-defined pre-existing nodes (e.g. daily DevStack build).
- Supported drivers for cloud providers:
  - static host,
  - OpenStack,
  - Kubernetes,
  - OpenShift,
  - AWS EC2.

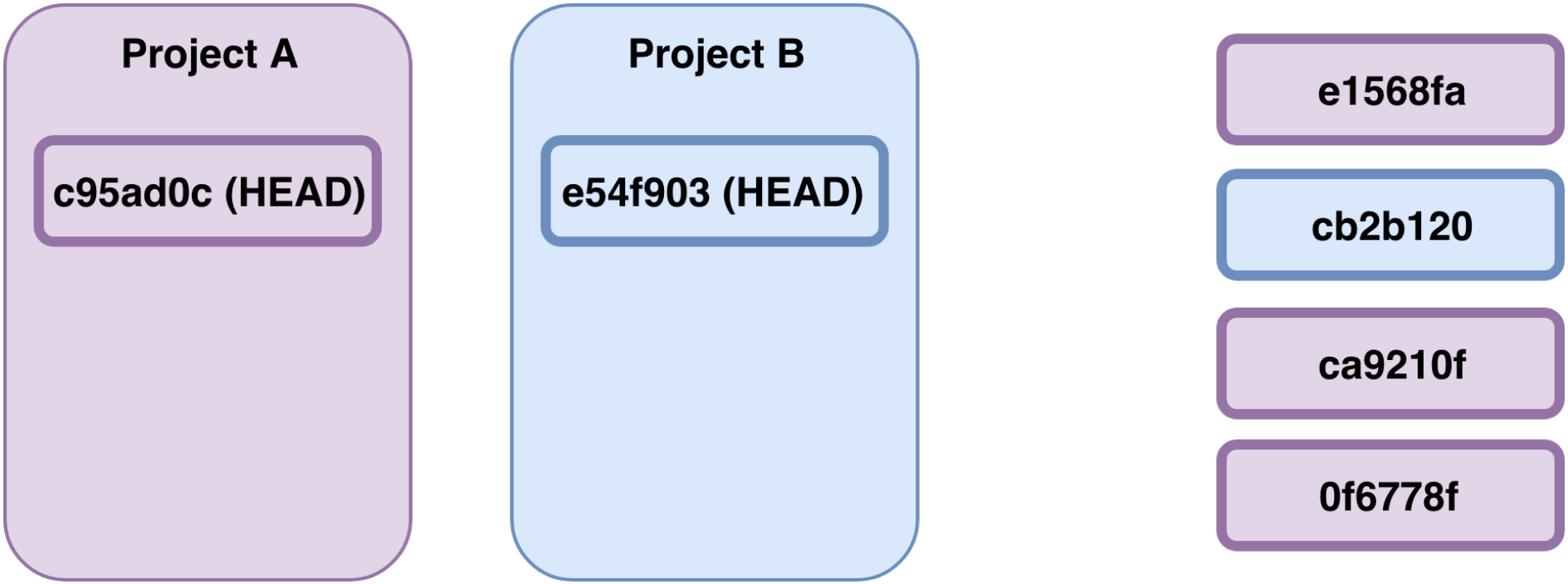
OVHcloud

How does Zuul work? (1/10)



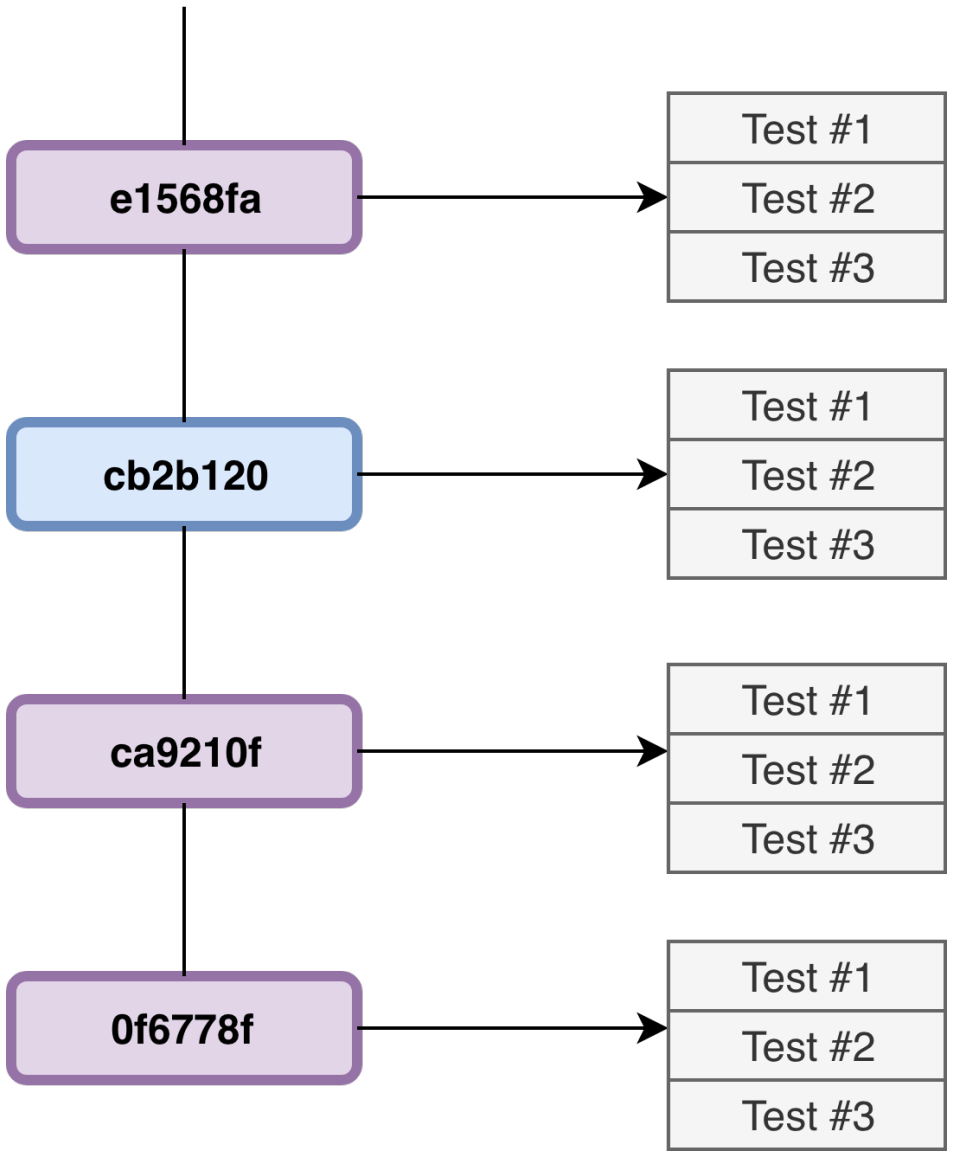
OVHcloud

How does Zuul work? (2/10)



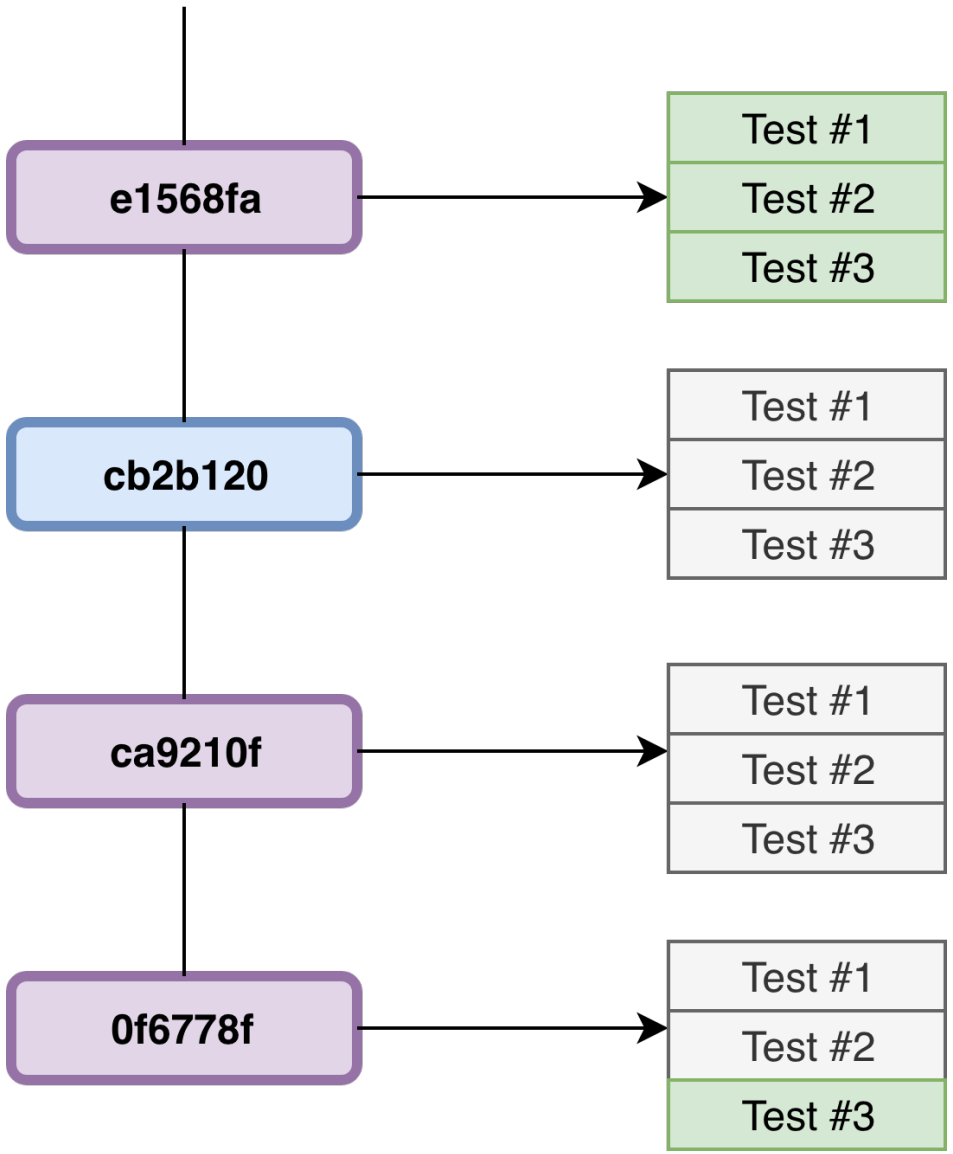
OVHcloud

How does Zuul work? (3/10)



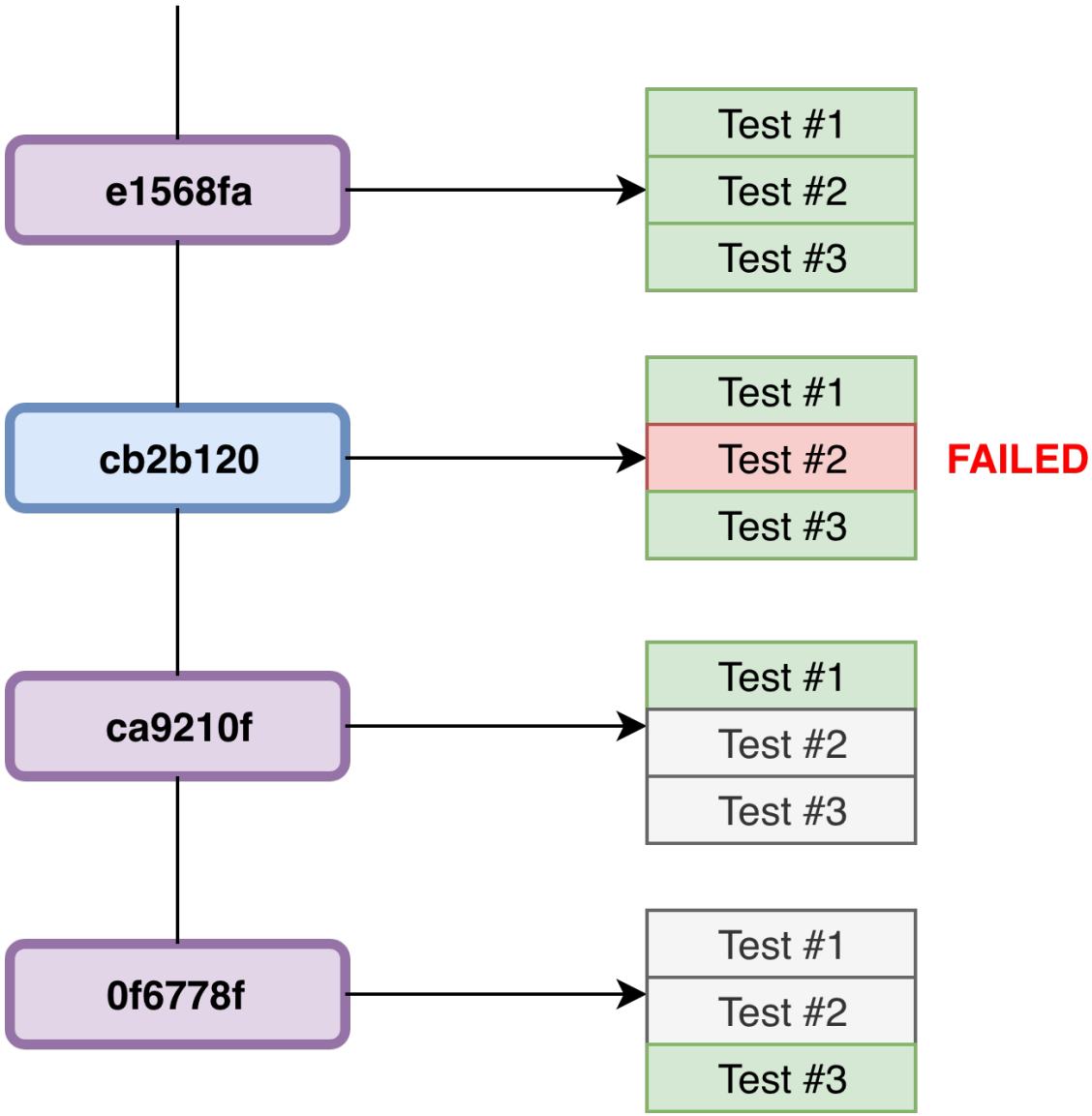
OVHcloud

How does Zuul work? (4/10)



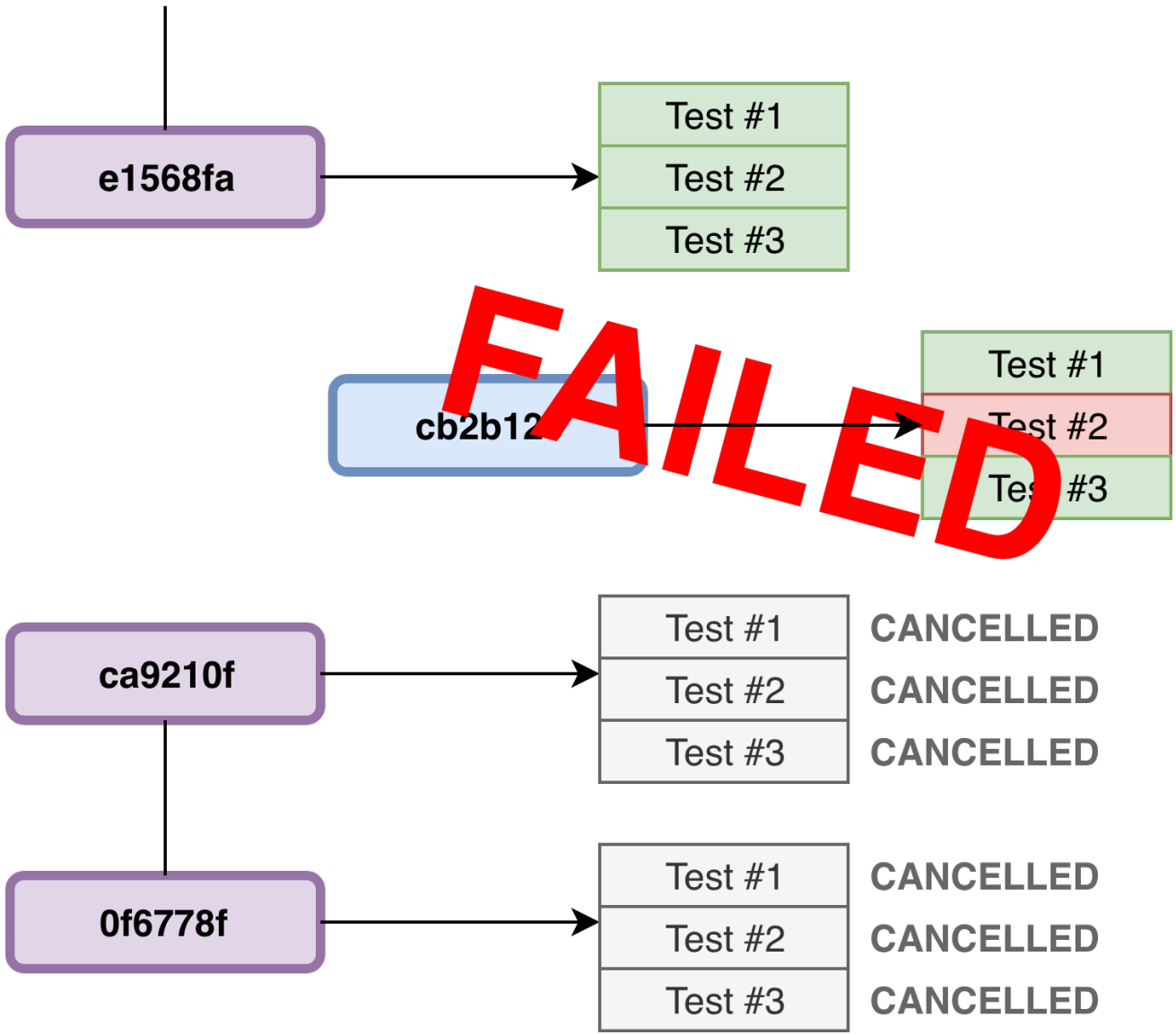
OVHcloud

How does Zuul work? (5/10)



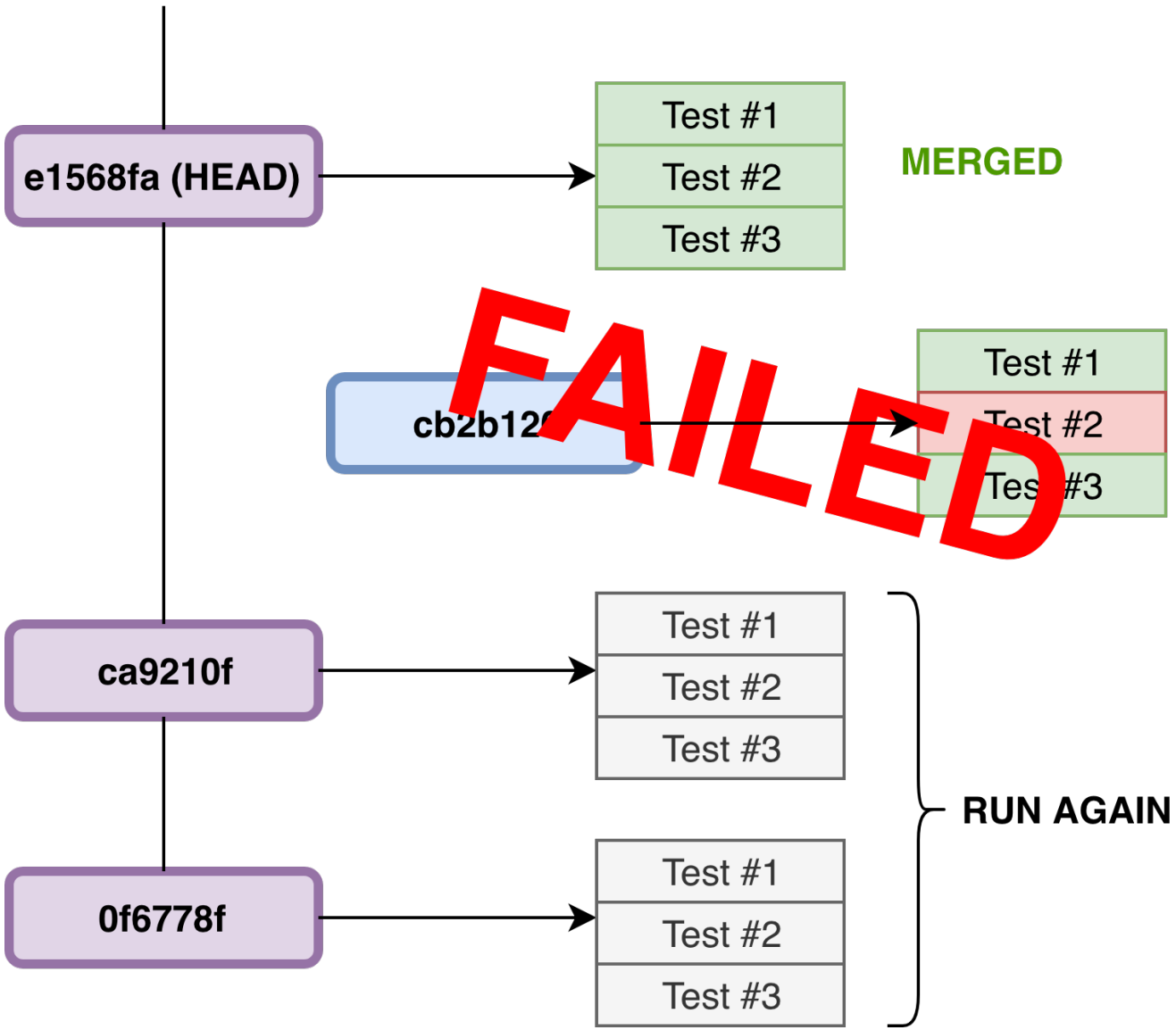
OVHcloud

How does Zuul work? (6/10)



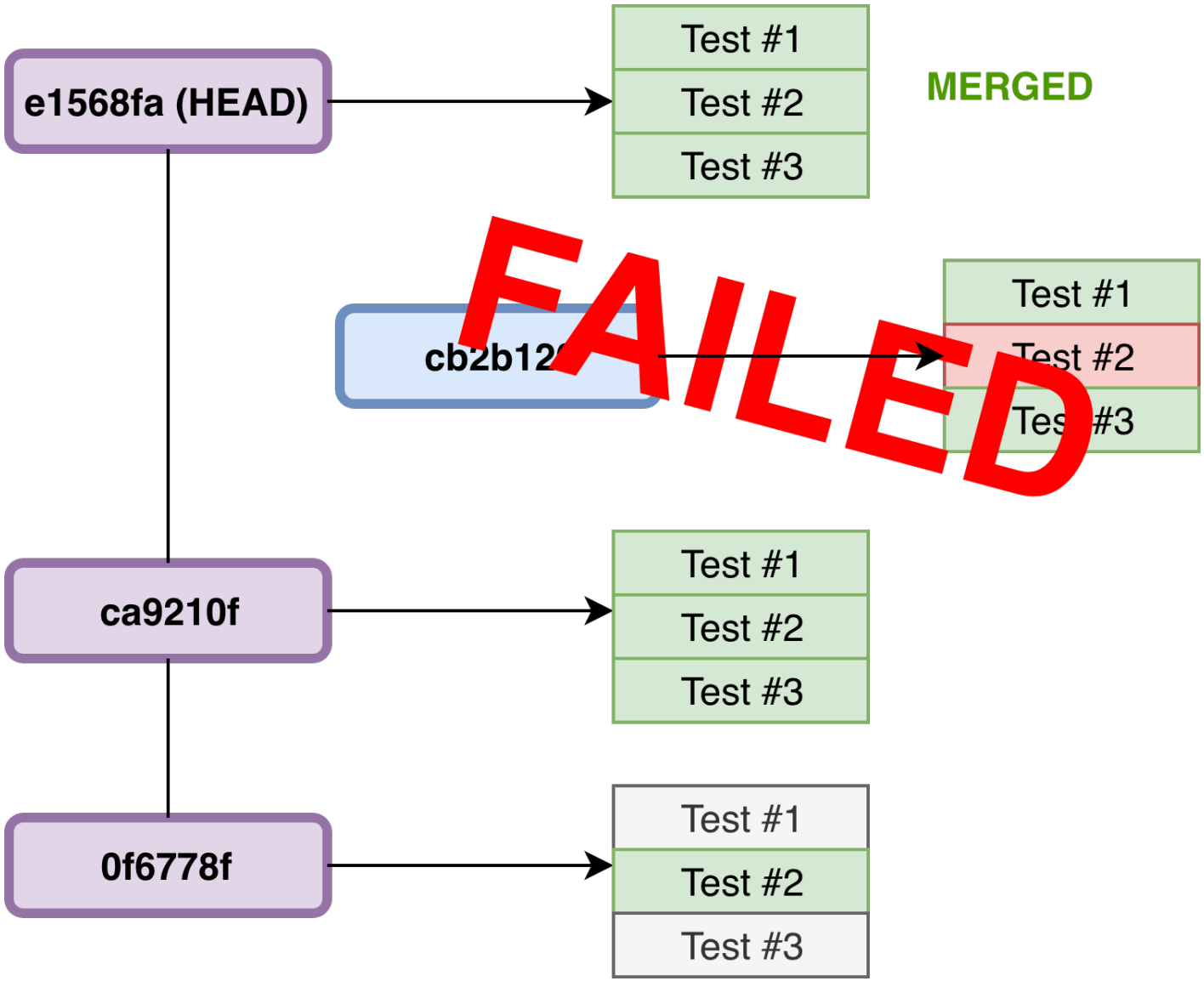
OVHcloud

How does Zuul work? (7/10)



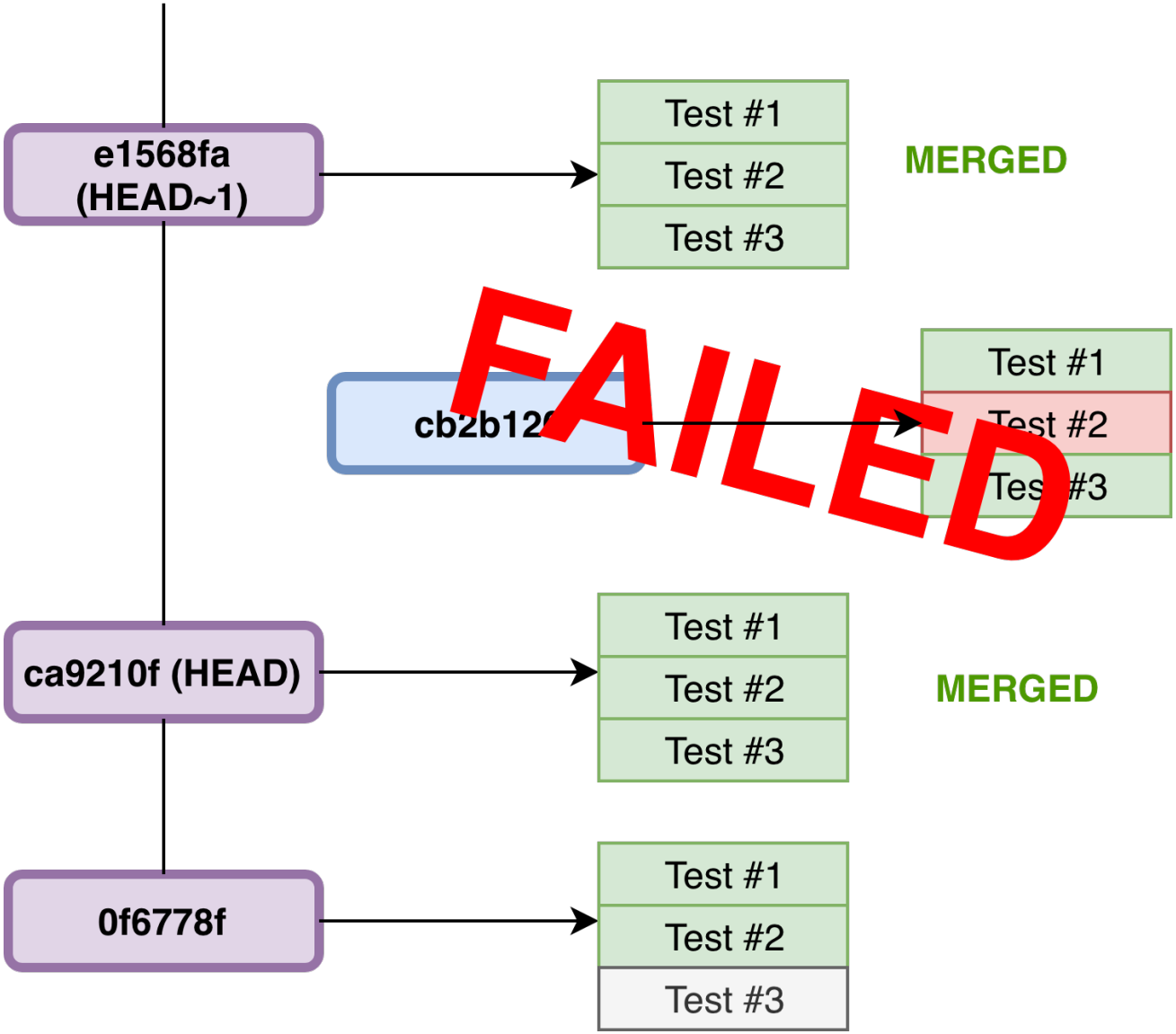
OVHcloud

How does Zuul work? (8/10)



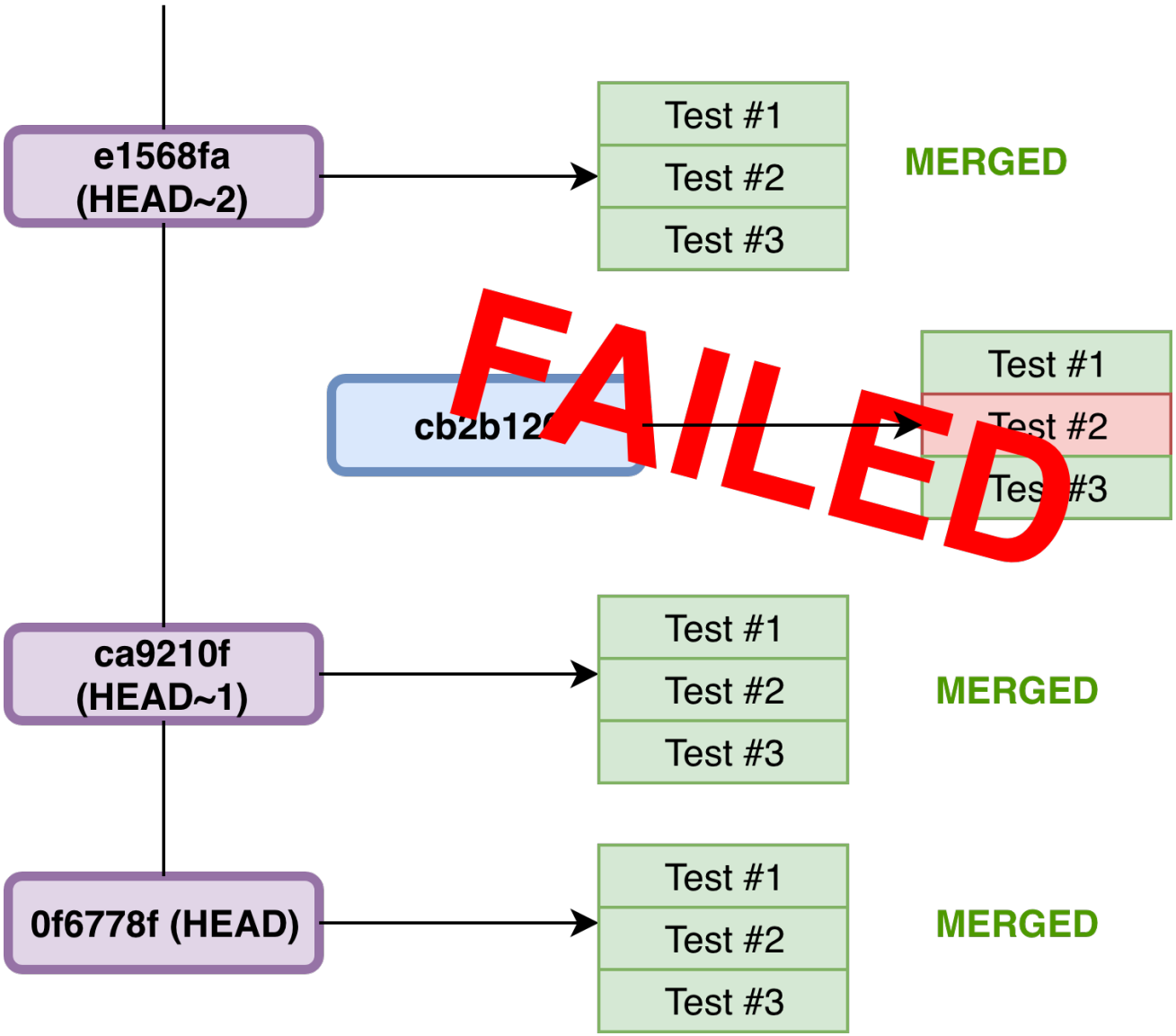
OVHcloud

How does Zuul work? (9/10)



OVHcloud

How does Zuul work? (10/10)



OVHcloud

Pipeline types

Type	Example case
check	on change upload (unit, integration tests), Verified ±1
gate	before a change is merged, last final verification
post	after change has been merged
periodic	once a while, every specified time
experimental	unstable tests, experimental features
silent	¯_(ツ)_/¯
pre-release	when a new pre-release tag uploded
release	when a changed is marked as a release

OVHcloud

Configuration (1/2)

Zuul recognizes two types of projects:

- config projects – adjust global behaviour, templates, etc.,
- untrusted projects – any regular repository to test.

Zuul will read its configuration from any of:

- `zuul.d/` directory,
- `.zuul.d/` directory,
- `zuul.yaml` file,
- `.zuul.yaml` file.

OVHcloud

Configuration (2/2)

```
#
# Example: https://github.com/openstack/nova/blob/master/.zuul.yaml
#
- job:
    name: nova-live-migration
    parent: nova-dsvm-multinode-base
    description: |
      Run tempest live migration tests against both local storage and (...)
    run: playbooks/legacy/nova-live-migration/run.yaml
    post-run: playbooks/legacy/nova-live-migration/post.yaml
    (...)
  (...)
- project:
    templates:
      - check-requirements
      - integrated-gate-py3
      (...)
    check:
      jobs:
        - ironic-tempest-ipa-wholedisk-bios-agent_ipmitool-tinyipa:
            voting: false
        - nova-grenade-live-migration
        - nova-live-migration
        (...)
```

OVHcloud

Quick start (1/4)

1. Install prerequisites

```
sudo apt-get install docker-compose git python3-pip
sudo python3 -m pip install git-review
```

2. Get the source code

```
git clone https://opendev.org/zuul/zuul
```

3. Start the containers

```
cd zuul/doc/source/admin/examples
sudo -E docker-compose up
```

Based on: https://zuul-ci.org/docs/zuul/admin/quick-start.html (12th June 2019).

OVHcloud

Quick start (2/4)

Results:

- started services:
  - Zookeeper,
  - Gerrit,
  - Nodepool,
  - Zuul Scheduler,
  - Zuul Web Server,
  - Zuul Executor,
  - Apache HTTPD.
- Gerrit API/SSH: `localhost:29418`
- Gerrit HTTP: `http://localhost:8080/`
- Zuul Web: `http://localhost:9000/`

OVHcloud

Quick start (3/4)

4. In Gerrit: add new user and configure it, e.g. set e-mail and SSH key.
5. Clone the **zuul-config** repo and set basic Zuul configuration (as user):

```
git clone http://localhost:8080/zuul-config
cd zuul-config && mkdir zuul.d
vim -p zuul.d/{jobs.yaml,pipelines.yaml,projects.yaml}
git add zuul.d && git commit -m 'init' && git review
```

6. In Gerrit: merge the new configuration (as admin).
7. Clone the **test1** repo and define new test job (as user):

```
git clone http://localhost:8080/test1
cd test1 && mkdir playbooks
vim -p playbooks/testjob.yaml .zuul.yaml
git add . && git commit -m 'basic tests' && git review
```

8. In Gerrit and Zuul: observe what happened :-)

OVHcloud

Quick start (4/4)

OVHcloud

Qucik start – supplement: jobs.yaml and projects.yaml

```yaml
#
# Example for quick start step 5:
# jobs.yaml
#
- job:
    name: base
    parent: null
    nodeset:
      nodes:
        - name: ubuntu-bionic
          label: ubuntu-bionic
```

```yaml
#
# Example for quick start step 5:
# projects.yaml
#
- project:
    name: ^.*$
    check:
      jobs: []
    gate:
      jobs: []

- project:
    name: zuul-config
    check:
      jobs:
        - noop
    gate:
      jobs:
        - noop
```

OVHcloud

Qucik start – supplement: pipelines.yaml (1/2)

```yaml
#
# Example for quick start step 5: pipelines.yaml
#
- pipeline:
    name: check
    description: |
      Newly uploaded patchsets enter this pipeline to receive an
      initial +/-1 Verified vote.
    manager: independent
    require:
      gerrit:
        open: True
        current-patchset: True
    trigger:
      gerrit:
        - event: patchset-created
        - event: change-restored
        - event: comment-added
          comment: (?i)^(Patch Set [0-9]+:)?( [\w\\+-]*)*(\n\n)?\s*recheck
    success:
      gerrit:
        Verified: 1
      mysql:
    failure:
      gerrit:
        Verified: -1
      mysql:

(...)
# Continuation on next slide
```

OVHcloud

Qucik start – supplement: pipelines.yaml (2/2)

```
31| (...)
32| - pipeline:
33|     name: gate
34|     description: |
35|       Changes that have been approved are enqueued in order in this
36|       pipeline, and if they pass tests, will be merged.
37|     manager: dependent
38|     post-review: True
39|     require:
40|       gerrit:
41|         open: True
42|         current-patchset: True
43|         approval:
44|           - Workflow: 1
45|     trigger:
46|       gerrit:
47|         - event: comment-added
48|           approval:
49|             - Workflow: 1
50|     start:
51|       gerrit:
52|         Verified: 0
53|     success:
54|       gerrit:
55|         Verified: 2
56|         submit: true
57|       mysql:
58|     failure:
59|       gerrit:
60|         Verified: -2
61|       mysql:
```

OVHcloud

Qucik start – supplement: testjob.yaml and .zuul.yaml

```yaml
#
# Example for quick start step 7: playbooks/testjob.yaml
#
- hosts: all
  tasks:
    - debug:
        msg: Hello world!
```

```yaml
#
# Example for quick start step 7: .zuul.yaml
#
- job:
    name: testjob
    run: playbooks/testjob.yaml

- project:
    check:
      jobs:
        - testjob
    gate:
      jobs:
        - testjob
```

OVHcloud

Zuul versus rest of the world...

Zuul:

- has much more complex architecture to set up,
- is specialized and oriented for the specific job,
- supports cross-project and cross-repository dependencies,
- keeps all the configuration in source code repository,
- reacts only to defined events in source code repository manager.

OVHcloud

Where to begin?

Official web site: `https://zuul-ci.org/`

- user: `https://zuul-ci.org/docs/zuul/user/index.html`
- admin: `https://zuul-ci.org/docs/zuul/admin/index.html`
- dev: `https://zuul-ci.org/docs/zuul/developer/index.html`

Packages:

- `https://pypi.org/project/zuul/`
- `https://pypi.org/project/nodepool/`

Source code:

- `https://opendev.org/zuul/zuul`
- `https://opendev.org/zuul/nodepool`

OVHcloud

Thank you for your attention!

The slides are available: `http://datko.pl/zuul.pdf`

Supplementary materials: `http://datko.pl/zuul.tgz`

OVHcloud

Zuul, the Third

Throws Away Any Dirt!

Szymon Datko
szymon.datko@corp.ovh.com

Roman Dobosz
rdobosz@redhat.com

6th November 2019